

CLAIMS

1 1. Method for verifying a transformer of a so-called source code into a
2 so-called transformed code designed for an embedded system, said source and
3 transformed codes being associated with virtual machines, characterized in that it
4 comprises at least the following steps:

5 - determining, for each of said source (1) and transformed (3) codes, a first
6 common subset (13), constituting a single virtual machine that factors in the behavior
7 of these two codes (1,3);

8 - determining, for each of said source (1) and transformed (3) codes, a second
9 subset (10, 30) constituted by a plurality of so-called auxiliary functions ($10_i - 30_i$)
10 used by said single virtual machine, said auxiliary functions ($10_i - 30_i$) representing
11 residual differences between said source (1) and transformed (3) codes;

12 - associating said auxiliary functions in pairs, a first auxiliary function (10_i) of
13 each pair belonging to said second subset (10) associated with said source code (1)
14 and a second auxiliary function (30_i) of each pair belonging to said second subset (30)
15 associated with said transformed code (3);

16 - verifying (6) a given correspondence property between said auxiliary
17 functions ($10_i - 30_i$) of all of said pairs; and

18 - verifying that said transformation of the source code (1) into a transformed
19 code (3) by said converter (2) satisfies said given correspondence property.

1 2. Method according to claim 1, characterized in that said correspondence
2 property is a logical relation, so that said auxiliary functions of each of said pairs (10_i
3 - 30_i), when executed, generate results linked by said logical relation, and in that this
4 relation is the identity relation for so-called observable entities of each of said source
5 and transformed codes, for any pair of auxiliary functions, so that the functionalities
6 of said source code (1) are retained when said transformation into said transformed
7 code (3), and said verification of the code transformer (2), are performed.

1 3. Application of the verification method according to either of claims 1
2 and 2 to a code transformer (2) that generates, from said source code (1), a
3 transformed code (3) designed to be stored in memory means (71) of a chip card (7).

1 4. Application according to claim 3, characterized in that, said
2 transformed code being a program written in the virtual machine of a given computer
3 language, said chip card (7) is a chip card that stores a plurality of software
4 applications (A_1 through A_n) written in this transformed code (3).

1 5. Application according to claim 3 or 4, characterized in that said source
2 code (1) is a program written in the "JAVA" (registered trademark) virtual machine
3 and said transformed code (3) is a program written in the "JAVA CARD" (registered
4 trademark) virtual machine.

Added
A7

0978614-060801